**LAB 1:**

**Title :Install and setup python environment**

**Installing Python**

Download and Install Python from the official website https://www.python.org/downloads/

**Find the Python Version & Installation Directory**

python --version

where python

**To Install python Packages :** pip install <<packagename>>

**To list installed python packages :** pip list

**Using a virtual environment**

**To install virtual environment:** pip install virtualenv

**To Check the version of virtual environment :** virtualenv –version

**To Create your own virtual environment**

virtualenv <env\_name>>

OR

python -m venv <<myenv>>

**Activate Virtual Environment on Windows :**

cd <env\_name>

<env\_name>\Scripts\activate

**Deactivate Virtual Environment**

deactivate <env\_name>

**LAB 4:**

**Title : Create a basic Flask application with a route that returns “Hello, World!” when accessed via the browser**

**Code:**

**app.py**

from flask import Flask,render\_template

app=Flask(\_\_name\_\_)

from flask\_mysqldb import MySQL

@app.route('/')

def base():

    return 'Hello,World'

if \_\_name\_\_=='\_\_main\_\_':

    app.run(debug=True)

**Output:**



**LAB 5:**

**Title: Develop a web form using HTML and Flask to capture user input (e.g., name, email) and display the input on another page.**

Code:

**register.html**

{% extends 'base.html' %}

 {% block title %}Register Page{% endblock%}

 {% block content %}

      <div class="col-md-4">

    <h1>User Registration</h1>

    <form method="post" action="/register">

      <label>First Name</label>

      <input type="text" required  class="form-control" name="first\_name" /> <br />

      <label>Last Name</label>

      <input type="text" class="form-control" name="last\_name" /> <br />

      <label>Email</label>

      <input type="email" class="form-control" name="email" /> <br />

      <label>Password</label>

      <input type="password" class="form-control" name="password" /> <br />

      <label>Confirm password</label>

      <input type="password" class="form-control" name="confirm\_password" /> <br />

      <button type="submit" class="btn btn-primary">Signup</button>

      <a href="/login" class="btn btn-secondary">Already have an account? Login Here</a>

    </form>

    </div>

 {% endblock%}

**display.html**

{% extends "base.html" %} {% block title %}Register Success{% endblock %} {%

block content %}

<div class="container mt-5">

  <h1>

    Hi, {{ first\_name }} {{ last\_name }}, hope you are fine! Your email is: {{

    email }}

  </h1>

</div>

{% endblock %}

**app.py**

@app.route("/register", methods=["GET","POST"])

def register\_page():

    if request.method=="POST":

        first\_name = request.form.get("first\_name")

        last\_name = request.form["last\_name"]

        email = request.form["email"]

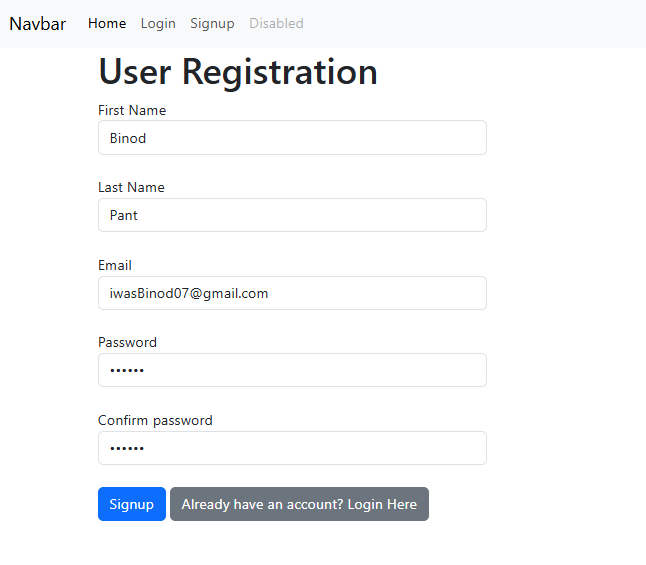
        password = request.form["password"]

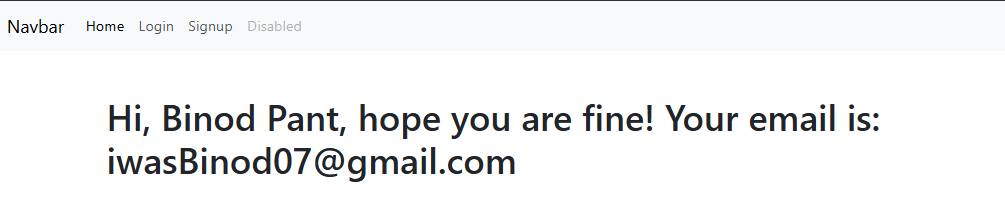
        confirm\_password = request.form["confirm\_password"]

        return render\_template("display.html",first\_name=first\_name, last\_name=last\_name, email=email)

    return render\_template("register.html")

**Output:**





**LAB 6:**

**Title: Write a Flask application that connects to a MySQL database using mysql-connector with Template integration.**

**Code:**

**app.py**

from flask import Flask, render\_template, request, redirect,url\_for, flash

from flask\_mysqldb import MySQL

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = 'your\_secret\_key'

# Configure MySQL

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = 'root'

app.config['MYSQL\_DB'] = 'flask\_class'

mysql = MySQL(app)

@app.route("/")

@app.route("/home")

def index\_page():

    cur = mysql.connection.cursor()

    cur.execute("SELECT \* FROM user")

    data = cur.fetchall()

    cur.close()

    return render\_template("home.html", users=data)

@app.route("/login", methods=["GET","POST"])

def login\_page():

    if request.method=="POST":

        username = request.form["username"]

        password = request.form["password"]

        if username=='nast' and password=='becomputer':

            return redirect(url\_for("index\_page"))

        else:

            flash("Username/Password not matched!","danger")

            return redirect(url\_for("login\_page"))

    return render\_template("login.html")

@app.route("/register", methods=["GET","POST"])

def register\_page():

    if request.method=="POST":

        first\_name = request.form.get("first\_name")

        last\_name = request.form["last\_name"]

        email = request.form["email"]

        password = request.form["password"]

        confirm\_password = request.form["confirm\_password"]

        #save to database

        cur = mysql.connection.cursor()

        cur.execute("INSERT INTO user(first\_name, last\_name, email, password, status) VALUES (%s, %s, %s, %s, %s)", (first\_name,last\_name, email, password,1))

        mysql.connection.commit()

        cur.close()

        return redirect("/")

    return render\_template("register.html")

if \_\_name\_\_=="\_\_main\_\_":

    app.run(debug=True)

**base.html**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=\, initial-scale=1.0" />

    <link

      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/css/bootstrap.min.css"

      rel="stylesheet"

      integrity="sha384-LN+7fdVzj6u52u30Kp6M/trliBMCMKTyK833zpbD+pXdCLuTusPj697FH4R/5mcr"

      crossorigin="anonymous"

    />

    <title>{% block title %} Base Title {% endblock%}</title>

  </head>

  <body>

    <nav class="navbar navbar-expand-lg bg-body-tertiary">

      <div class="container-fluid">

        <a class="navbar-brand" href="/">Navbar</a>

        <button

          class="navbar-toggler"

          type="button"

          data-bs-toggle="collapse"

          data-bs-target="#navbarNav"

          aria-controls="navbarNav"

          aria-expanded="false"

          aria-label="Toggle navigation"

        >

          <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

          <ul class="navbar-nav">

            <li class="nav-item">

              <a class="nav-link active" aria-current="page" href="/">Home</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="/login">Login</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="/register">Signup</a>

            </li>

            <li class="nav-item">

              <a class="nav-link disabled" aria-disabled="true">Disabled</a>

            </li>

          </ul>

        </div>

      </div>

    </nav>

    <div class="container">{% block content %} {% endblock%}</div>

    <script

      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/js/bootstrap.bundle.min.js"

      integrity="sha384-ndDqU0Gzau9qJ1lfW4pNLlhNTkCfHzAVBReH9diLvGRem5+R9g2FzA8ZGN954O5Q"

      crossorigin="anonymous"

    ></script>

  </body>

</html>

**register.html**

{% extends 'base.html' %}

 {% block title %}Register Page{% endblock%}

 {% block content %}

      <div class="col-md-4">

    <h1>User Registration</h1>

    <form method="post" action="/register">

      <label>First Name</label>

      <input type="text" required  class="form-control" name="first\_name" /> <br />

      <label>Last Name</label>

      <input type="text" class="form-control" name="last\_name" /> <br />

      <label>Email</label>

      <input type="email" class="form-control" name="email" /> <br />

      <label>Password</label>

      <input type="password" class="form-control" name="password" /> <br />

      <label>Confirm password</label>

      <input type="password" class="form-control" name="confirm\_password" /> <br />

      <button type="submit" class="btn btn-primary">Signup</button>

      <a href="/login" class="btn btn-secondary">Already have an account? Login Here</a>

    </form>

    </div>

 {% endblock%}

Home.html

{% extends 'base.html' %}

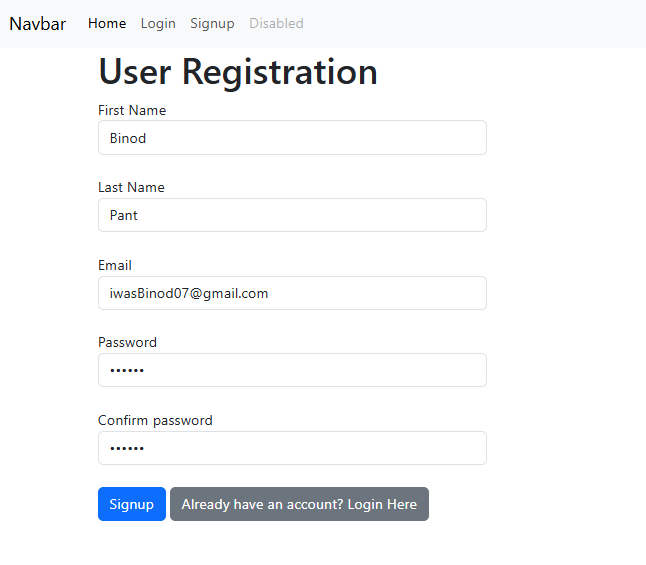
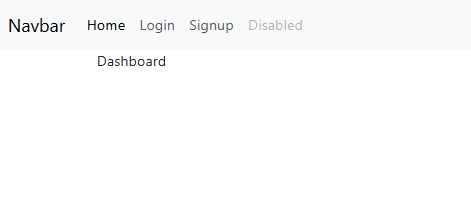
 {% block title %}Dashboard Page{% endblock%}

 {% block content %}

Dashboard

 {% endblock %}

**Output:**



**Lab 7:**

**LAB 7:**

**Title: Write a Flask application that connects to a MySQL database using SQLAlchemy. Create a table called students with columns id, name, age, and grade. Insert records and fetch data from the database to display on a webpage.**

**Code :**

**app.py**

from flask import Flask,render\_template,request,url\_for,redirect

from flask\_sqlalchemy import SQLAlchemy

app=Flask(\_\_name\_\_)

# mysql connection  ( using py mysql sriver)

app.config['SQLALCHEMY\_DATABASE\_URI']='mysql+pymysql://root:root@localhost/flask\_crud'

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS']=False

db=SQLAlchemy(app)

# Model

class Student(db.Model):

    id=db.Column(db.Integer,primary\_key=True)

    name=db.Column(db.String(100),nullable=False)

    age=db.Column(db.Integer,nullable=False)

    grade=db.Column(db.String(200),nullable=False)

@app.route('/')

def index\_page():

    data=students=Student.query.all()

    return render\_template('index.html',students=data)

@app.route('/create',methods=['GET','POST'])

def create():

    if request.method=='POST':

        name= request.form['name']

        age= request.form['age']

        grade= request.form['grade']

        new\_student=Student(name=name,age=age,grade=grade)

        db.session.add(new\_student)

        db.session.commit()

        return redirect(url\_for('index\_page'))

    return render\_template('create.html')

@app.route('/update/<int:id>',methods=['GET','POST'])

def update(id):

    student=Student.query.get\_or\_404(id)

    if request.method=="POST":

        student.name= request.form['name']

        student.age= request.form['age']

        student.grade= request.form['grade']

        db.session.commit()

        return redirect(url\_for('index.html'))

    return render\_template('update.html',student=student)

@app.route('/delete/<int:id>')

def delete(id):

    student=Student.query.get\_or\_404(id)

    db.session.delete(student)

    db.session.commit()

    return redirect(url\_for('index.html'))

if \_\_name\_\_=="\_\_main\_\_":

    with app.app\_context():

        db.create\_all()

    app.run(debug=True)

c**reate.html**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Add student</title>

  </head>

  <body>

    <h2>Add student</h2>

    <form method="POST" action="/create">

      Name:<input type="text" name="name" required /><br />

      Age:<input type="number" name="age" required /><br />

      Grade:<input type="text" name="grade" required /><br />

      <input type="submit" value="Save" />

    </form>

    <a href="{{url\_for('index\_page')}}">Back</a>

  </body>

</html>

**Index.html**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Student list</title>

  </head>

  <body>

    <h2>Student list</h2>

    <a href="/create">Add Student</a>

    <table border="" width="100%">

      <tr>

        <th>ID</th>

        <th>Name</th>

        <th>Age</th>

        <th>Grade</th>

        <th>Actions</th>

      </tr>

      {% for s in students %}

      <tr>

        <td>{{s.id}}</td>

        <td>{{s.name}}</td>

        <td>{{s.age}}</td>

        <td>{{s.grade}}</td>

        <td>

          <a href="{{ url\_for('update', id=s.id) }}">Edit</a> |

          <a href="{{ url\_for('delete', id=s.id) }}">Delete</a> l̥

        </td>

      </tr>

      {% endfor %}

    </table>

  </body>

</html>

**update.html**

<!DOCTYPE html>

<html>

  <head>

    <title>Edit Student</title>

  </head>

  <body>

    <h2>Edit Student</h2>

    <form method="POST">

      Name:<input

        type="text"

        name="name"

        value="{{student.name}}"

        required

      /><br />

      Age:<input

        type="number"

        name="age"

        value="{{student.age}}"

        required

      /><br />

      Grade:<input

        type="text"

        name="grade"

        value="{{student.grade}}"

        required

      /><br />

      <input type="submit" value="Update" />

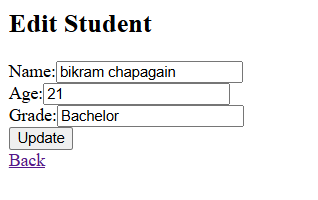
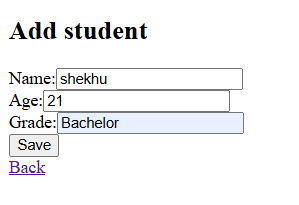
    </form>

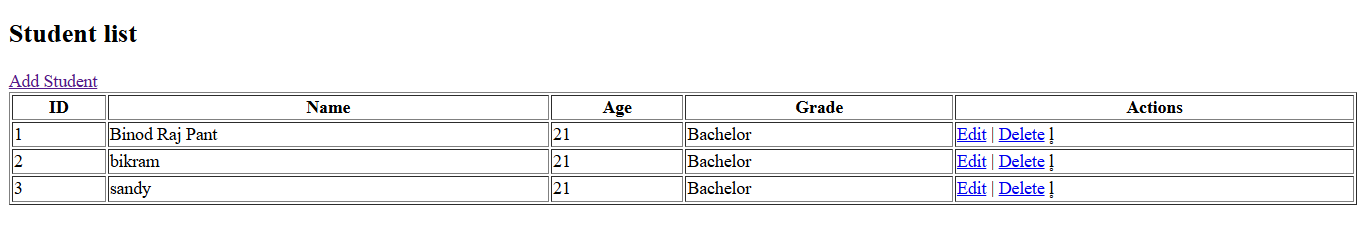
    <a href="{{ url\_for('index\_page') }}">Back</a>

  </body>

</html>

**Output:**





**LAB 8:**

**Title: Implement HTML drop-down to display dynamic options. Dynamic HTML tables product list and product details using flask.**

**Code :**

**app.py**

class Category(db.Model):

    id=db.Column(db.Integer,primary\_key=True)

    code=db.Column(db.String(20),nullable=False)

    name=db.Column(db.String(200),nullable=False)

    description=db.Column(db.String(500),nullable=False)

    product=relationship('Product' , back\_populates="category")

class Product(db.Model):

    id=db.Column(db.Integer,primary\_key=True)

    category\_id=db.Column(db.Integer,db.ForeignKey('category.id'))

    name=db.Column(db.String(100),nullable=False)

    quantity=db.Column(db.Numeric(10,2),nullable=False)

    rate=db.Column(db.Numeric(10,2),nullable=False)

    unit\_of\_measurement=db.Column(db.String(200),nullable=False)

    category=relationship('Category',back\_populates='product')

@app.route('/add\_Category',methods=['POST','GET'])

def add\_category():

    if request.method=='POST':

        code=request.form.get('code')

        name=request.form.get('name')

        description=request.form.get('description')

        category=Category(code=code,name=name,description=description)

        db.session.add(category)

        db.session.commit()

    return render\_template('add\_category.html')

@app.route('/add\_product',methods=['POST','GET'])

def add\_product():

    if request.method=='POST':

        category=request.form.get('category\_id')

        name=request.form.get('name')

        qyt=request.form.get('qyt')

        rate=request.form.get('rate')

        unit=request.form.get('unit')

        product=Product(category\_id=category,name=name,quantity=qyt,rate=rate,unit\_of\_measurement=unit)

        db.session.add(product)

        db.session.commit()

    category=Category.query.all()

    return render\_template('add\_product.html',category=category)

**base.html**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>{%block title%} {%endblock%}</title>

    <link

      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/css/bootstrap.min.css"

      rel="stylesheet"

      integrity="sha384-LN+7fdVzj6u52u30Kp6M/trliBMCMKTyK833zpbD+pXdCLuTusPj697FH4R/5mcr"

      crossorigin="anonymous"

    />

  </head>

  <body>

    <nav class="navbar navbar-expand-lg bg-body-tertiary">

      <div class="container-fluid">

        <a class="navbar-brand" href="{{url\_for('index\_page')}}">CRUD</a>

        <button

          class="navbar-toggler"

          type="button"

          data-bs-toggle="collapse"

          data-bs-target="#navbarNav"

          aria-controls="navbarNav"

          aria-expanded="false"

          aria-label="Toggle navigation"

        >

          <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

          <ul class="navbar-nav">

            <li class="nav-item">

              <a

                class="nav-link active"

                aria-current="page"

                href="{{url\_for('index\_page')}}"

                >Home</a

              >

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{{url\_for('create')}}">Create</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{{url\_for('add\_category')}}"

                >Add Category</a

              >

            </li>

            <li class="nav-item">

              <a class="nav-link" href="{{url\_for('add\_product')}}"

                >Add Product</a

              >

            </li>

          </ul>

        </div>

      </div>

    </nav>

    <div class="container">{%block content%} {%endblock%}</div>

    <script

      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.7/dist/js/bootstrap.bundle.min.js"

      integrity="sha384-ndDqU0Gzau9qJ1lfW4pNLlhNTkCfHzAVBReH9diLvGRem5+R9g2FzA8ZGN954O5Q"

      crossorigin="anonymous"

    ></script>

    >

  </body>

</html>

**add\_product.html**

{% extends 'base.html'%} {%block title%}add product{%endblock%} {%block

content%}

<h1>Add Product</h1>

<form action="{{url\_for('add\_product')}}" method="post">

  <label for="" class="form-label">Category:</label>

  <select class="form-select" name="category\_id">

    <option selected>Select Category</option>

    {%for c in category%}

    <option value="{{c.id}}">{{c.name}}</option>

    {%endfor%}

  </select>

  <label for="" class="form-label">name:</label>

  <input type="text" class="form form-control" name="name" /><br />

  <label for="">Qyt</label>

  <input type="number" class="form form-control" name="qyt" /><br />

  <label for="">Rate</label><br />

  <input type="number" class="form form-control" name="rate" /><br />

  <label for="">Unit of Measurement</label>

  <input type="text" class="form form-control" name="unit" /><br />

  <input

    type="submit"

    class="btn btn-success"

    name="measurement"

    value="Create"

  />

</form>

{%endblock%}

**add\_category.html**

{%extends 'base.html'%} {%block title%}add Catogery{%endblock%} {%block

content%}

<h1>Add Catogery</h1>

<form action="{{url\_for('add\_category')}}" method="post">

  <label for="" class="form-label">Category Code:</label>

  <input type="text" class="form form-control" name="code" id="" /><br />

  <label for="">Name</label>

  <input type="text" class="form form-control" name="name" id="" /><br />

<label for="">Discription</label><br />

  <textarea

    name="description"

    id=""

    class="form-control"

    placeholder="Leave a Discription here"

  ></textarea>

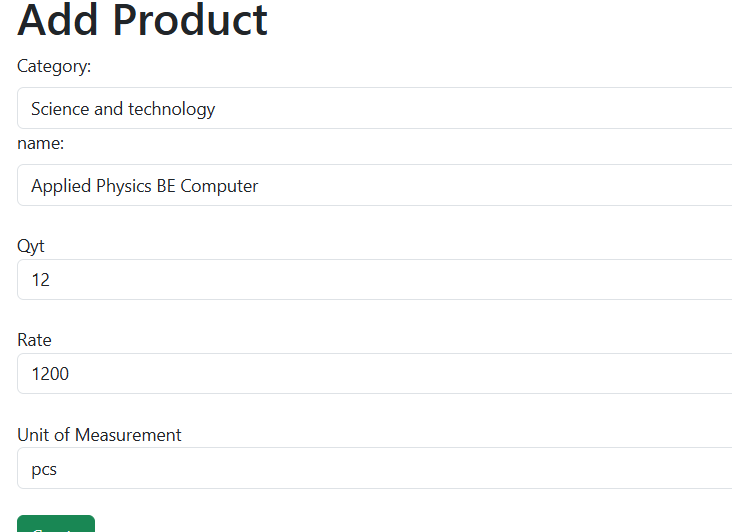
  <br />

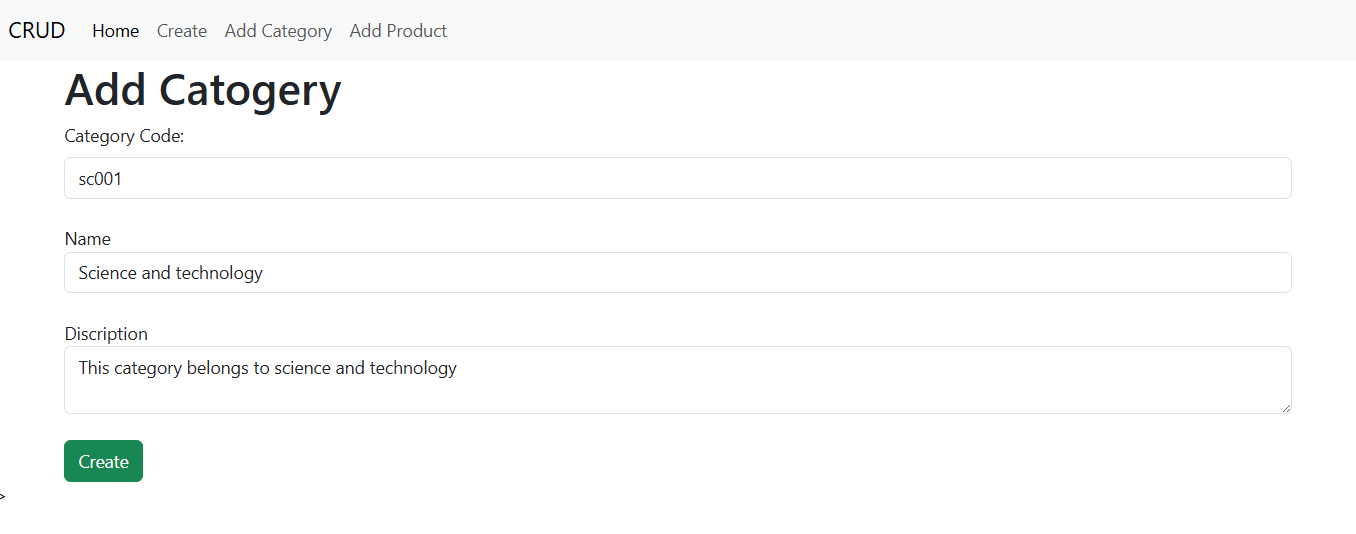
  <input type="submit" class="btn btn-success" value="Create" />

</form>

{%endblock%}

**Output:**





**LAB 9:**

**Title: Implementing REST API in Flask.**

**Code :**

**models.py**

from flask\_sqlalchemy import SQLAlchemy

db = SQLAlchemy()

class Student(db.Model):

id = db.Column(db.Integer, primary\_key=True)

name = db.Column(db.String(100), nullable=False)

age = db.Column(db.Integer, nullable=False)

email = db.Column(db.String(120), unique=True, nullable=False)

def to\_dict(self):

return {"id": self.id, "name": self.name, "age": self.age,

"email": self.email}

**app.py**

from flask import Flask, request, jsonify

from models import db, Student

app = Flask(\_\_name\_\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///database.db'

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db.init\_app(app)

@app.route('/')

def index():

return jsonify({"message": "Welcome to API!"})

# Create a student

@app.route('/students', methods=['POST'])

def create\_student():

data = request.get\_json()

new\_student = Student(name=data['name'], age=data['age'], email=data['email'])

db.session.add(new\_student)

db.session.commit()

return jsonify(new\_student.to\_dict()), 201

# Get all students

@app.route('/students', methods=['GET'])

def get\_students():

students = Student.query.all()

return jsonify([s.to\_dict() for s in students])

# Get a single student

@app.route('/students/<int:id>', methods=['GET'])

def get\_student(id):

student = Student.query.get(id)

if student:

return jsonify(student.to\_dict())

return jsonify({"message": "Student not found"}), 404

# Update a student

@app.route('/students/<int:id>', methods=['PUT'])

def update\_student(id):

student = Student.query.get(id)

if not student:

return jsonify({"message": "Student not found"}), 404

data = request.get\_json()

student.name = data.get('name', student.name)

student.age = data.get('age', student.age)

student.email = data.get('email', student.email)

db.session.commit()

return jsonify(student.to\_dict())

# Delete a student

@app.route('/students/<int:id>', methods=['DELETE'])

def delete\_student(id):

student = Student.query.get(id)

if not student:

return jsonify({"message": "Student not found"}), 404

db.session.delete(student)

db.session.commit()

return jsonify({'message': 'Student deleted successfully'})

if \_\_name\_\_ == '\_\_main\_\_':

with app.app\_context():

db.create\_all()

app.run(debug=True)

**LAB 9:**

**Title: Develop a login page using flask that store information in a db password must be in hashed form.**

**Code :**

**app.py**

from flask import Flask, render\_template, request, redirect, url\_for, session, flash

from flask\_sqlalchemy import SQLAlchemy

from werkzeug.security import generate\_password\_hash, check\_password\_hash

import os

app = Flask(\_\_name\_\_)

app.secret\_key = 'your\_secret\_key\_here'  # Replace with a secure key

# SQLite DB setup

basedir = os.path.abspath(os.path.dirname(\_\_file\_\_))

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///' + os.path.join(basedir, 'users.db')

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db = SQLAlchemy(app)

# User model

class User(db.Model):

    id = db.Column(db.Integer, primary\_key=True)

    username = db.Column(db.String(150), unique=True, nullable=False)

    password = db.Column(db.String(200), nullable=False)

# Create the database

with app.app\_context():

    db.create\_all()

@app.route('/')

def home():

    if 'user' in session:

        return render\_template('home.html', username=session['user'])

    return redirect(url\_for('login'))

@app.route('/register', methods=['GET', 'POST'])

def register():

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        existing\_user = User.query.filter\_by(username=username).first()

        if existing\_user:

            flash('Username already exists.')

            return redirect(url\_for('register'))

        hashed\_password = generate\_password\_hash(password)

        new\_user = User(username=username, password=hashed\_password)

        db.session.add(new\_user)

        db.session.commit()

        flash('Registration successful! Please log in.')

        return redirect(url\_for('login'))

    return render\_template('register.html')

@app.route('/login', methods=['GET', 'POST'])

def login():

    if request.method == 'POST':

        username = request.form['username']

        password = request.form['password']

        user = User.query.filter\_by(username=username).first()

        if user and check\_password\_hash(user.password, password):

            session['user'] = user.username

            return redirect(url\_for('home'))

        else:

            flash('Invalid credentials')

            return redirect(url\_for('login'))

    return render\_template('login.html')

@app.route('/logout')

def logout():

    session.pop('user', None)

    flash('Logged out successfully!')

    return redirect(url\_for('login'))

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**home.html**

<!doctype html>

<html>

<head><title>Home</title></head>

<body>

    <h2>Welcome, {{ username }}!</h2>

    <a href="{{ url\_for('logout') }}">Logout</a>

</body>

</html>

**login.html**

<!doctype html>

<html>

<head><title>Login</title></head>

<body>

    <h2>Login</h2>

    <form method="post">

        Username: <input type="text" name="username" required><br>

        Password: <input type="password" name="password" required><br>

        <input type="submit" value="Login">

    </form>

    <p>Don't have an account? <a href="{{ url\_for('register') }}">Register here</a></p>

    {% with messages = get\_flashed\_messages() %}

      {% if messages %}

        <ul>{% for msg in messages %}<li>{{ msg }}</li>{% endfor %}</ul>

      {% endif %}

    {% endwith %}

</body>

</html>

**register.html**

<!doctype html>

<html>

<head><title>Register</title></head>

<body>

    <h2>Register</h2>

    <form method="post">

        Username: <input type="text" name="username" required><br>

        Password: <input type="password" name="password" required><br>

        <input type="submit" value="Register">

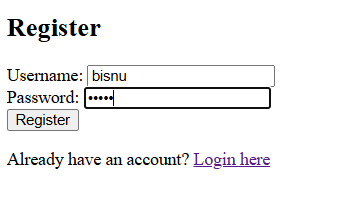
    </form>

    <p>Already have an account? <a href="{{ url\_for('login') }}">Login here</a></p>

    {% with messages = get\_flashed\_messages() %}

      {% if messages %}

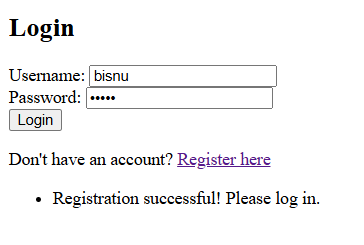
        <ul>{% for msg in messages %}<li>{{ msg }}</li>{% endfor %}</ul>

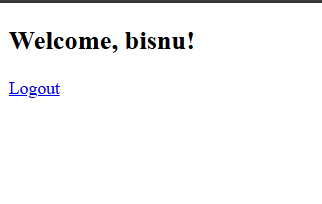
      {% endif %}

    {% endwith %}

</body>

</html>





**LAB 10:**

**Title: Sending an Email using Flask.**

**Code :**

**app.py**

from flask import Flask, render\_template, request, redirect, url\_for, flash

from flask\_mail import Mail, Message

app = Flask(\_\_name\_\_)

app.secret\_key = 'applefgtrtr'

# Email Configuration

app.config['MAIL\_SERVER'] = 'smtp.gmail.com'

app.config['MAIL\_PORT'] = 587

app.config['MAIL\_USE\_TLS'] = True

app.config['MAIL\_USERNAME'] = 'www.mingmaarchy568@gmail.com'

app.config['MAIL\_PASSWORD'] = 'qvov deae uvtr geei'  # Consider using environment variables for security

mail = Mail(app)

@app.route("/")

def index():

    return render\_template("index.html")

@app.route("/send-email", methods=["GET", "POST"])

def send\_email():

    if request.method == "POST":

        try:

            email = request.form['email']

            subject = request.form['subject']

            message = request.form['message']

            msg = Message(subject,

                          sender=app.config['MAIL\_USERNAME'],

                          recipients=[email])

            msg.body = message

            mail.send(msg)

            flash('Email Sent Successfully!', 'success')

        except Exception as e:

            flash(f'Error Sending Email: {str(e)}', 'danger')

        return redirect(url\_for('index'))

    return render\_template("send-email.html")

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

**index.html**

{% extends 'base.html' %} {% block title %}Send Email Page{% endblock %} {%

block content %}

<div class="container mt-5">

  <div class="row justify-content-center">

    <div class="col-md-6">

      <h3 class="mb-4 text-center">Send Email</h3>

      <form action="/send-email" method="post">

        <div class="form-floating mb-3">

          <input

            type="email"

            name="email"

            class="form-control"

            id="floatingTo"

            placeholder="Recipient's Email"

            required

          />

          <label for="floatingTo">To</label>

        </div>

        <div class="form-floating mb-3">

          <input

            type="text"

            name="subject"

            class="form-control"

            id="floatingSubject"

            placeholder="Subject"

            required

          />

          <label for="floatingSubject">Subject</label>

        </div>

        <div class="form-floating mb-3">

          <textarea

            class="form-control"

            name="message"

            id="floatingMessage"

            placeholder="Message"

            style="height: 150px"

            required

          ></textarea>

          <label for="floatingMessage">Message</label>

        </div>

        <button type="submit" class="btn btn-primary w-100">Send</button>

      </form>

      <!-- Flash Messages -->

      {% with messages = get\_flashed\_messages(with\_categories=true) %} {% if

      messages %}

      <div class="mt-4">

        {% for category, message in messages %}

        <div

          class="alert alert-{{ category }} alert-dismissible fade show"

          role="alert"

        >

          {{ message }}

          <button

            type="button"

            class="btn-close"

            data-bs-dismiss="alert"

            aria-label="Close"

          ></button>

        </div>

        {% endfor %}

      </div>

      {% endif %} {% endwith %}

    </div>

  </div>

</div>

{% endblock %}

**Output:**

